Predicting Fraud in Mobile Money Transfer Using Case-Based Reasoning

Adeyinka Adedoyin[1], Stelios Kapetanakis[1], Georgios Samakovitis[2], and Miltos Petridis[3]

[1] University of Brighton, Brighton, UK
{A.Adedoyin, S.Kapetanakis}@brighton.ac.uk
[2] University of Greenwich, London, UK
G.Samakovitis@gre.ac.uk
[3] Middlesex University, London, UK
M.Petridis@mdx.ac.uk

Abstract. This paper proposes an improved CBR approach for the identification of money transfer fraud in Mobile Money Transfer (MMT) environments. Standard CBR capability is augmented by machine learning techniques to assign parameter weights in the sample dataset and automate k-value random selection in k-NN classification to improve CBR performance. The CBR system observes users' transaction behaviour within the MMT service and tries to detect abnormal patterns in the transaction flows. To capture user behaviour effectively, the CBR system classifies the log information into five contexts and then combines them into a single dimension, instead of using the conventional approach where the transaction amount, time dimensions or features dimension are used individually. The applicability of the proposed augmented CBR system is evaluated using simulation data. From the results, both dimensions show good performance with the context of information weighted CBR system outperforming the individual features approach.

Keywords: Money transfer fraud, Case-based reasoning, Genetic algorithm, Simulation data, Mobile money.

1 Introduction

Mobile Money Transfer (MMT) services are financial services provided by a Mobile Network Operator (MNO) that enable transfer of funds using a digital equivalent of cash (electronic money) between service subscribers through mobile channels [1]. While in developed countries MMT is merely seen as an extension to existing banking services, several developing countries, where access to banking is often challenging for individuals and businesses, tend to view mobile money transfer technologies as platforms with significant strategic and societal value in supporting financial inclusion to unbanked and under-banked populations. More than 2.5 billion adults globally lack a formal bank account, with the majority in developing countries. However, approximately 68 percent of that population

have access to a mobile phone [2]. In a 2013 Gartner report [3], the worldwide market for MMT was estimated to reach over 450 million subscribers in 2017, with a mobile transaction value of more than $721 billion. The main drives behind the success of mobile money are the explosive growth in the number of mobile devices and the drop in computing power cost, which has made mobile phones more accessible [4].

The ability of MMT to handle large numbers of small value payments, its suitability for transferring funds worldwide in digital currencies, and the current absence of robust regulatory oversight, makes it both an attractive target for attackers and fraudsters, and an equally attractive vehicle for money laundering [5]. While in most countries Anti-Money Laundering (AML) and transaction fraud reporting is compulsory for service providers and financial institutions [1], in many of them, existing ML legislation is not presently fit to fully accommodate the relatively young m-money markets. This absence of suitable oversight intensifies the risk exposure of MMT to fraud, money laundering and other financial misuse. For example, where proper controls are not deployed, fraudsters can get access to MMT services without disclosing their identity to the MNO, by taking advantage of prepaid phones, pooling and delegation of mobile devices [1, 6].

A crucial observation is made at this point to distinguish between capabilities for investigating transaction fraud, as opposed to these addressing the identification of money laundering; while transaction fraud is typically recognised as most commonly associated with money laundering [1], money laundering activity itself may technically exist in the absence of transaction fraud (e.g. through the use of mule accounts [1]). Even more crucially, money laundering is $process-driven$, as opposed to transaction fraud, which is $event-driven$. As a consequence, AML predictive modelling is far more complex and computationally demanding than fraud monitoring, while selection of suitable Artificial Intelligence approaches becomes significantly more challenging for AML. In the context of this work we are considering the development of monitoring and predictive models for transaction fraud, and with view to merely supporting AML indirectly.

Different types of monitoring and predictive models have been proposed for identifying fraud in financial transactions streams [1]. Most are based on data-driven (machine learning) methods, typically requiring a significant amount of financial transaction historical data [7]. The challenges in obtaining real life financial transaction data sets for research purposes are well-known [8] including data protection and confidentiality, ethical issues, time, and the cost associated with collecting multiple instances of a diverse set of data sources. In addition, when real life data sets are available, these may be small in size and lack information on confirmed fraud cases and their possible taxonomies [9]. A case-based reasoning approach offers an alternative that is commensurate with the limited datasets described above. It is more transparent than black-box models, such as neural networks and has the ability to operate with limited experience, learn and improve predictive accuracy as more data becomes available [7, 10]. To the

best of our knowledge, there is relatively limited literature on applying CBR to the field of financial transaction fraud detection.

In this paper, we propose an improved CBR approach by complementing standard CBR methods with machine learning capabilities for assigning parameter weights and automating the random selection of k-value in order to detect financial transaction fraud. Both the standard and proposed CBR approach were analysed using simulation dataset. We motivate the use of the proposed CBR approach by comparing it's results with that of the standard CBR.

The remaining parts of this paper are organized as follows: Section 2 presents a short overview of related work on financial fraud detection using Case-based reasoning. Section 3 discusses our CBR system approach. In Section 4, our experiment data and implementation are presented, and in Section 5 this approach is empirically evaluated using simulated data generated with Multi-agent based simulator in [11]. Finally, Section 6 concludes the paper and outlines future works.

2 Background

Research from the literature on predicting fraud in financial transaction services has focused on statistical, machine learning and other classification techniques and they all provide effective results. However, the design of statistical and rule-based system requires a significant amount of expert knowledge which, in turn, makes the process costly and time-consuming.

As an alternative, machine learning methods such as Artificial Neural Networks (ANN), Support Vector Machines (SVM) and Bayesian belief network have been widely used to predict different types of financial transaction fraud following a data-driven approach (i.e on the basis of past observations of fraudulent / genuine transactions). For example, Bekirev et al. [12] and Mohamed et al. [13] used a feedforward approach to detect payment card fraud and telecommunication fraud respectively. In addition, the literature reports hybrid approaches, where statistical techniques are combined with neural networks to predict financial fraud. Ravisankar et al. [14] used a probabilistic neural network to identify companies that resort to financial statement fraud. Examples of prediction methods based on neural network and SVMs include [15]. However in the absence of significant size of historical data, they tend not to perform well. A detailed review of machine learning applications in solving financial fraud problems is provided by Albashrawi in [16].

Case-based Reasoning method as an alternative to standard machine learning methods, comes with a number of advantages when applied to the field of financial transaction fraud. For example, Case-based Reasoning features has the ability to (i) learn in the absence of historical consumption data, while continuously improving when more data becomes available over time, (ii) realize knowledge transfer as spending habits evolve; as is the case where information on one transaction is exploited to improve predictions for different yet similar transactions, and (iii) provide precedent-based justification instead of justify-

ing a solution by showing a trace of the rules that led to decision [17, 18]. One of the initial works where CBR approach was applied to the field of financial transaction fraud was published by Cheol-Soo and Ingoo [19]. They used multi-agent Case-based reasoning approach to reduce the number of final-line fraud investigation in credit approval process, achieving precise results.

In [20] and [21], promising results were produced with a simplified CBR model for monitoring and predicting financial transaction fraud. However, the predictive accuracy of that model was lower than that of a neural network of similar complexity and featured a relatively high false positive rate. As discussed in [22], this identified weakness is considered as damaging as high false negative rates for customer trust, acutely reflecting why precision requirements for operational fraud detection systems are high, and partly explaining current reluctance to adopt unified industry-wide approaches. This paper therefore seeks to deliver improved performance by supplementing the standard CBR capabilities by using a machine learning technique to assign parameter weights in the sample dataset and automate k-value random selection in k-NN classification between the range 3,5,7 or 9.

3 CBR Model

This section describes our CBR system and the feature weights optimization, followed by a brief outline of the dataset used. The section concludes with a discussion of the experiments and their results.

3.1 Case Representation

The use of mobile money transfer varies widely across households due to a number of aspects of consumer behaviour like the product and brand choice, purchase amount, and income group [23]. This indicates that consumer purchase behaviour is temporal in nature. Thus, as the spending behaviour of customers is temporal in nature and most of the individuals exhibit consistent spending behaviour, an event-driven process chain of transactions can be a robust representation of the spending behaviour. Therefore, in order to represent the behavioural pattern of users, it is necessary to define events that model the MMT process. However, according to [24], it is challenging to derive a workflow of transactions from the control flow of mobile money systems, because every user is free to use the system as they wish (for instance, the user can choose their own amounts, frequencies, communities of interests, etc.). For this reason our events representation was generated from the users behaviour in the mobile money system. For each process instance, there is pair of active users and type of transaction (i.e., (user1,CASHIN)), making the assumption that the amounts in transactions of the same type (i.e., only CASHIN, only TRANSFER, etc.) are similar, while amounts of different types of transaction are not [24, 25]. For the needs of event representation and case construction in the transaction streams, five different types of mobile-enabled financial operations available in the sample data were

used, namely: Money Deposit (A), Money Withdrawal (B), Merchant Payment (C), Person-to-person transfer (D), and Airtime Recharge (E). A possible graphical representation of user's behaviour in the log trace can be seen in Fig. 1.

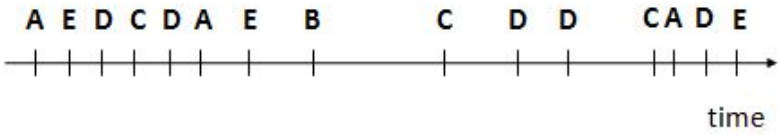

Fig. 1: Possible representation of user's behaviour

In mobile money transfer transaction processing, the spending behaviour contains information about the transaction amount, time gap since last transaction, day of the week, etc. The transaction amount, frequency, and time are closely related to spending behaviour of a person which are actually influenced by income, resource availability, and lifestyle of the person. In most conventional fraud detection system (FDS), the transaction amount is considered as the most important parameter for fraud detection. Also, previous research work has shown that efficiency of any FDS is associated with the amount and time dimensions separately [26]. However, we propose the classification of information into five contexts and combine them into a single dimension to capture user behaviour effectively. This gives significant improvement in accuracy over a system that considers each feature dimension individually [26]. Below are the five types of information context used:

1. **Transaction type:** Transaction type entities
2. **Client:** Features of entities (client ID, Profile e.g savings or current account, account balance, spending habit category)
3. **Interval:** Features of entities (Month of the year, Day of the Week).
4. **Location:** Location entities
5. **Amount**: Quantization of amount entities into finite levels up to maximum daily spending limit.

For the CBR system process representation, we start with a simple definition. Consider the total set of events E in the log file, each event is a quintuple $(\overline{v_i})$ representing the different contexts of information for the events.

$$\overline{v_i} = [\, c_1, c_2, \ldots, c_n \,] \tag{1}$$

where (c_n) is the context of type n, $(n = 1 \ldots 5)$

Each instance of query (Z_m) can be defined as:

$$Z_m = [\ \overline{v_1^z}, \ldots, \overline{v_m^z}\ ] \tag{2}$$

where m is the length of the event query Z_m.

For the Case base representation, each case (X) contains a description (D) with the corresponding solutions (S), i.e an outcome tag (y) associated to each event or transaction type. That is,

$$X = [\ D_x, S_x\ ] \tag{3}$$

where

$$D_x = [\ \overline{v_1^x}, \ldots, \overline{v_5^x}\ ] \tag{4}$$

$$S_x = y_1, \ldots, y_n \tag{5}$$

However, to classify the transaction outcome our system uses a binary classification (safe, and fraudulent), therefore $n = 2$.

3.2 Case Similarity

For the needs of similarity measure, the similarity between two cases is defined as a weighted average of the vector similarities. In previous work, the flexibility of weighting was not exploited, i.e all weights were simply set to the same value. However, since different vectors are obviously of different importance, we decided to take advantage of the capability of genetic algorithms to determine the optimal weights for each vector. During the retrieval process, an ordered list of k most similar cases to the query were retrieved and returned. This was implemented using a k-Nearest Neighbour algorithm. The overall similarity value was computed by weighting the local similarity of each vector ($\overline{v}_i$). The resulting value is weighted with a value (w_i) that represents the relevance of the corresponding transaction in the global similarity computation:

$$Sim(Z_m, D_x) = \sum_{j=1}^{m} w_j * \sigma(\overline{v_j^z}, \overline{v_j^x}) \tag{6}$$

where m is the length of the event query and $\sigma(\overline{v_j^z}, \overline{v_j^x})$ is the similarity between the jth events in the target query and source case in the case base respectively.

3.3 CBR Model Weights

In order to exploit the flexibility of weighting all the input vectors, a Genetic Algorithm (henceforth GA) was used to calculate their weight so as to reflect the significance of each vector as determined by the GA procedure. Optimal weighting of variables using GA was extensively used in the literature, as for instance in [27–29]. The method in [27] was adapted for the configuration of GA in obtaining the optimal weights for each of the vectors. In the experiment,

the GA uses a population of individuals representing the different weights, and the generation of the population evolves until the individual weights with the best performance is returned. Each individual weight contains both the vector weight and k parameter of the K-Nearest Neighbour algorithm to estimate the best number of cases that must be retrieved to classify new transactions. For the needs of configuration, the genetic algorithm was run with an initial population of 1000 individuals. Each individual contains the weights of each vector and the value of k (a random value 3, 5, 7 or 9) that the CBR model uses in the retrieval stage. Also, at the initial stage a random value was assigned to each weight, then later normalised to sum of 1. The following cycle is repeated until there is no more improvement in the performance of the best individual population:

1. Fitness Evaluation: At this stage the genetic algorithm executes a cross-validation of the weights and k value for each individual population to generate the fitness performance.
2. Remove: After the evaluation of all the individual population, 25% of the population with the worst fitness performance was removed.
3. Cross-over: To reproduce the population that was removed (i.e 25% individual removed after the fitness evaluation), the genetic algorithm combines the individual population with the best performance. During the cross-over process, the parent individuals are taken in pairs and then combined together to form a new individual called child. The weight of each child individual contains the average weights of the parents (normalised weight) and the value of k is computed analogously.
4. Mutation: During the implementation of the mutation function, the Individuals along with their weights are chosen randomly for modification, using 5% of the population. These modification prevents local maximum values.

4 Experiment

4.1 Data

Obtaining transaction data from financial institutions can be difficult due to a number of reasons, such as ethical limitations, privacy issues and government or corporate policies. In addition, when such data are made available they may be small in quantity, lack information on confirmed fraud cases, or carry limited features information. Therefore, for the evaluation of our suggested approach we simulated transactions using a Multi-Agent Simulation Tool-kit (MASON) [30] that combines the behaviour and habits of several users within a Mobile money environment. Multi-agent based simulators have been extensively used in the literature to represent agents with different behaviours in a swarm, as in [11, 31–33]. In this paper, the simulator was built according to the methodology proposed in [34] and implemented using adapted multi-agent based simulator (MABS) developed by Lopez et al. [11].

Simulation walk-through: The first step of the simulation was to set up agents and their locations. Then different clients that will be present in the simulation

were randomly generated, and each client is assigned an ID. A client state at each time depends on a Markov transition matrix that assigns when to change from Active to Inactive and from savings to current account, with higher limits on daily transaction. The clients in this simulation have basic operations; they can either make a deposit, withdrawal, person-to-person transfer, pay a merchant, buy airtime or decide not to perform any transaction. If a client needs to perform an action, it conducts a local search within its network to see which of its neighbours are in active state. If the search is successful, then it places a request for a type of operation using a probabilistic transition function. The request placed depends on the transition function from client account balance, daily limits on each clients account type, and user spending habits category. When the balance is high the agent has a higher probability to make a withdrawal, transfer, pay a merchant, or airtime recharge, rather than a deposit. Fig. 2 outlines these activities.

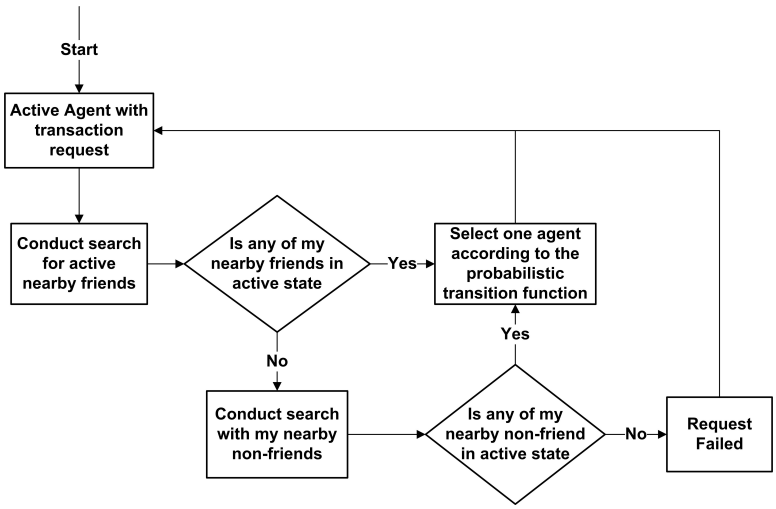

Fig. 2: A flowchart representing the simulation Walk-through

However, if the search is unsuccessful, the client can delegate a request to an inactive client to conduct a local search within its own neighbourhood for a mediator. Once this is achieved a routing record is created with information about the originator of the request. At each pass, the routing record is updated with information about the intermediate requestor. At some point, if the search is successful then the delegated client places a request to perform an operation on behalf of the initial requesting client. The delegation of request stops after a search is conducted in the neighbourhood a level above the requesting clients level. For each simulation the input parameters values were modified in order the improve the quality of the simulation. A total of 2000 end users were created from different cities performing several transactions with partners either inside or

outside of their network. The simulator stores transactions details in a log file and each entry contains informations such as the transaction type, amount, sender and receiver profile (Id, account type), time-stamp etc. The simulation was run for six months between 1^{st} of September 2013 and 28^{th} of February 2014. At the end of the data preparation phase a total of 141,556 transactions were generated with 282 transactions (0.2%) labelled as suspicious. The data generated by the simulation represent a realistic situation of common class imbalance problem in financial transaction dataset, where one of the classes is very large in comparison to the other one.

4.2 Experiment Implementation

The CBR approach was implemented as described in the previous section using jCOLIBRI framework [35]; a Java framework that allows rapid prototyping of a CBR system, the development and deployment of the CBR system in real scenarios. For the needs of experiment evaluation, we compared our approach with the conventional individual feature similarity dimension as the baseline. In order to achieve this, two experiments were performed:

In the first experiment, labelled StdCBR (Standard CBR), the flexibility of weighting individual features was not exploited. All weights were assumed to be normalized and of equal importance for both feature dimensions i.e equal for individual feature dimension and classification of features in to five contexts. The global similarity measure was computed using Euclidean distance:

$$d = |Z - X| = \sqrt{\sum_{i=1}^{m} w_i |Z_i - X_i|^2} \tag{7}$$

where w_i is the weight of vector (attribute) i, Z is the query (new case), X is the source (retrieved case), m is the number of vectors in each case, and i is an individual vector from 1 to m.

As a potential improvement in the second experiment a featured weighted CBR system was carried out (FWCBR); relevant optimal weights for the features were assigned using a genetic algorithm and the value of k was computed analogously for both feature dimensions. The similarity function was computed by optimally combining the individual local similarity of $(\overline{v}_i)$ into a global similarity as discussed in section 3.2. For the needs of experiment evaluation due to the computational cost of genetic algorithm, the number of transaction data used from the simulation dataset were limited to 2000 out of which 0.084% were labelled as fraudulent. Only 25 users who had done more than 60 transactions (average transaction generated by the end users) were randomly selected. The experiments were ran for 10 iterations and the average was taken for the final classification result. For better precision, 5-fold cross validation was used.

5 Experimental Results (Evaluation)

In this section we present prediction performance results for both models FWCBR and StdCBR with K= 3. Use of larger values of k in the experiment did not present any significantly different results. The results associated to each model are shown in Tables I to IV. The standard CBR model (StdCBR) without weight optimization is used as a baseline for understanding the ground necessary for a conventional CBR system to solve the classification problem. The model shows extremely good results for a weighted context of information dimension (Table 3) with recall and precision levels of approximately 93% and 86% respectively. Although the standard CBR model with individual feature dimension shows the capability of detecting the positive class (78%), it features a low recall value (0.46%).

Table 1: Combined confusion matrix of the models from Individual feature dimension

Individual-Dimension				
	Fraud		**Normal**	
Prediction	**Std-CBR**	**FW-CBR**	**Std-CBR**	**FW-CBR**
Fraud	77	124	90	43
Normal	22	19	1811	1814

Table 2: Combined confusion matrix of the models from Context of information dimension

Context-Dimension				
	Fraud		**Normal**	
Prediction	**Std-CBR**	**FW-CBR**	**Std-CBR**	**FW-CBR**
Fraud	130	155	37	12
Normal	14	25	1819	1808

Table 3: Model comparison based on Recall and Precision in alignment with context

Model	**Individual-Dimension**		**Context-Dimension**	
	Recall	**Precision**	**Recall**	**Precision**
Std-CBR	0.46	0.78	0.74	0.87
FW-CBR	0.78	0.90	0.93	0.86

Table 4 describes the average accuracy for both models using the two different feature dimensions.The performance of all the four models exceeds 90%, with the weighted CBR system (FW-CBR) based on the context of information perspective leading (98%), as shown in Table 4.

Table 4: Average prediction accuracy of the models

Model	Individual Dimension	Context Dimension
StdCBR	0.94	0.96
FWCBR	0.97	0.98

Fig. 3 shows results from the developed CBR system. From the interface, 3 nearest neighbours can be seen for each new case, classification score (fraud as 1 and non-fruad as 0), as well as their similarity performance. This can provide a good insight into a number of final line case investigation for experts after the existing detection system has been utilised.

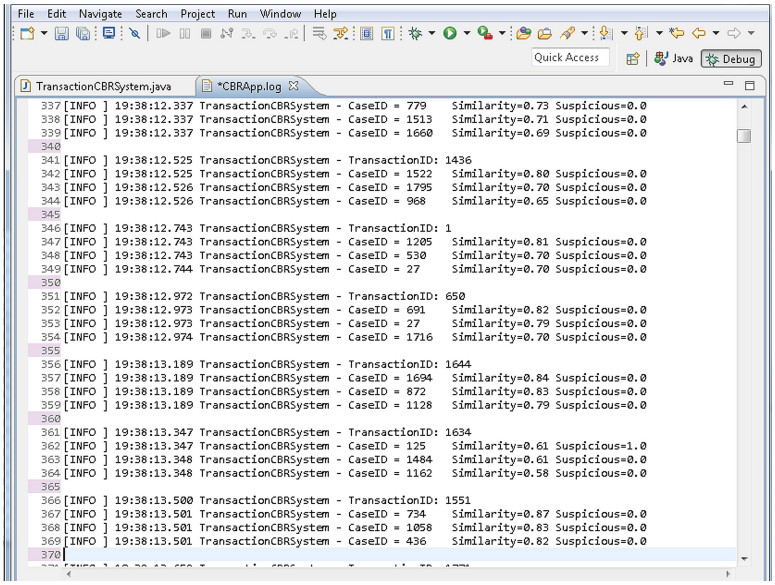

Fig. 3: Transaction neighbours summary

6 Conclusions

In this paper, an enhanced CBR model is proposed with the aim of improving the performance of standard CBR systems for fraud identification in Mobile Money Transfer (MMT). The enhanced system uses a combination of CBR and GA as a tool to optimize the significance level (weights) of the features. For the evaluation, instead of using the conventional approach where the transaction amount, time dimensions or features dimension are used individually, we classify the log information from the simulation data into five contexts and then combine them into a single dimension. Results demonstrate that the classification of log information into five contexts improves the performance of our proposed weighted CBR system with prediction accuracy of 0.97% and 0.98% for the two feature dimension perspectives. In addition, the ranking of clusters of transaction neighbours for new cases in the summary window may operate as an effective tool for experts to develop preliminary insight into suspicious transactions which can then be investigated in more detail. The computational complexity associated with the use of genetic algorithms is seen as one of the major challenges in our approach and more emphasis is placed in future work on reducing computation cost to improve the scalability of our proposed system.

References

1. Zhdanova, M., Repp, J., Rieke, R., Gaber, C., Hemery, B.: No smurfs: Revealing fraud chains in mobile money transfers. In: 9th International Conference on Availability, Reliability and Security, ARES 2014. (2014)
2. International Telecommunication Union: The Mobile Money Revolution Part 2: Financial Inclusion Enabler. ITU-T Technology Watch Report (2013)
3. Shen, S.: Forecast: Mobile Payment, Worldwide, 2013 Update (2013)
4. International Telecommunication Union: The Mobile Money Revolution. Part 1: NFC Mobile Payments. ITU-T Technology Watch Report (2013)
5. Bennett, N., Dilloway, S.: Investigating the Convergence of Money Laundering and Terrorist Financing. In: ACAMS AML and Financial Crime Conference. (2013)
6. Chatain, P.L., Zerzan, A., Noor, W., Dannaoui, N., de Koker, L.: Protecting Mobile Money against Financial Crimes: Global Policy Challenges and Solutions. The International Bank for Reconstruction and Development / The World Bank (2011)
7. Shabani, A., Paul, A., Platon, R., Hüllermeier, E.: Predicting the Electricity Consumption of Buildings: An Improved CBR Approach. In: 24th International Conference on Case-base reasoning (ICCBR 2016). (2016)
8. Bolton, R.J., Hand, D.J.: Statistical Fraud Detection A Review. Statistical Science (2002)
9. Gorton, D.: IncidentResponseSim: An AgentBased Simulation Tool for Risk Management of Online Fraud. In: Buchegger S., Dam M. (eds) Secure IT Systems, LNCS, Springer. (2015)
10. Platon, R., Dehkordi, V.R., Martel, J.: Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. Energy and Buildings, Elsevier (2015)

11. Lopez-rojas, E.A., Axelsson, S.: Multi Agent Based Simulation (MABS) of Financial Transactions for Anti Money Laundering (AML). In: 17th Nordic Conference on Secure IT. (2012)
12. Bekirev, A.S., Klimov, V.V., Kuzin, M.V., Shchukin, B.A.: Payment card fraud detection using neural network committee and clustering. Optical Memory and Neural Networks (Information Optics) (2015)
13. Mohamed, A., Bandi, A.F.M., Tamrin, A.R., Jaafar, M.D., Hasan, S., Jusof, F.: Telecommunication fraud prediction using backpropagation neural network (SoCPaR). In: International Conference of Soft Computing and Pattern Recognition, Malaysia (2009)
14. Ravisankar, P., Ravi, V., Raghava Rao, G., Bose, I.: Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Systems, ScienceDirect (2011)
15. Roselina, S., Subariah, I., Azlan, M.Z., Abdikarim, H.E.: Detecting SIM Box Fraud by Using Support Vector Machine and Artificial Neural Network. Jurnal Teknologi (2015)
16. Albashrawi, M., Lowell, M.: Detecting financial fraud using data mining techniques: A Decade Review from 2004 to 2015. Journal of Data Science (2016)
17. Chi, R.T., Kiang, M.Y.: An integrated approach of rule-based and case-based reasoning for decision support. In: Proceedings of the 19th annual conference on Computer Science (CSC '91). (1991)
18. Watson, I.: Case-based reasoning is a methodology not a technology. Knowledge-Based Systems, Elsevier (1999)
19. Park, C.S., Han, I.: A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction. Expert Systems with Applications, Elsevier (2002)
20. Adedoyin, A., Kapetanakis, S., Petridis, M., Panaousis, E.: Evaluating Case-Based Reasoning Knowledge Discovery in Fraud Detection. In: 24th Workshop on Case Based Reasoning (ICCBR2016): Synergies between CBR and Knowledge Discovery. (2016)
21. Kapetanakis, S., Samakovitis, G., Gunasekera, P.V.G.B., Petridis, M.: Monitoring Financial Transaction Fraud with the use of Case-based Reasoning. In: Seventeenth UK Workshop on Case-Based Reasoning. (2012)
22. Samakovitis, G., Kapetanakis, S.: Computer-aided Financial Fraud Detection: Promise and Applicability in Monitoring Financial Transaction Fraud. In: Proceedings of International Conference on Business Management and IS, Dubai, United Arab Emirates. (2013)
23. Slocum, J.W., Mathews, H.L.: Social Class and Income as Indicators of Consumer Credit Behavior. Journal of Marketing (2017)
24. Rieke, R., Zhdanova, M., Repp, J., Giot, R., Gaber, C.: Fraud Detection in Mobile Payments Utilizing Process Behavior Analysis. In: International Conference on Availability, Reliability and Security (ARES). (2013)
25. Giot, R., Gaber, C.: Predictive Security Analysis - Concepts, Implementation, first Results in Industrial Scenario (2013)
26. Kundu, A., Panigrahi, S., Sural, S., Majumdar, A.: BLAST-SSAHA Hybridization for Credit Card Fraud Detection. In: IEEE Transactions on Dependable and Secure Computing. (2009)
27. Jorro-Aragoneses, J.L., Díaz-agudo, B., Recio-garcía, J.A.: CBR tagging of emotions from facial expressions. In: Lamontagne L., Plaza E. (eds) Case-Based Reasoning Research and Development (ICCBR), Springer. (2014)

28. Manzoor, J., Asif, S., Masud, M., Khan, M.J.: Automatic Case Generation for Case-Based Reasoning Systems Using Genetic Algorithms. In: Third Global Congress on Intelligent Systems. (2012)
29. Ahn, H., Kim, K., Han, I.: Hybrid Genetic Algorithms and Case-Based Reasoning Systems. In: Computational and Information Science (CIS 2004). LNCS, Springer. (2004)
30. Luke, S., Cioffi-Reevilla, C., Panait, L., Sullivan, K., Cioffi-Revilla, C., Sullivan, K., Panait, L., Balan, G.: Mason: A multiagent simulation environment. Simulation (2005)
31. Tahir, A., Adeyinka, A.: Autonomic Service Management in Mobile Cloud Infrastructures. International Journal New Computure Architucture and their Application (2014)
32. Gaber, C., Hemery, B., Achemlal, M., Pasquet, M., Urien, P.: Synthetic logs generator for fraud detection in mobile transfer services. In: Sadeghi AR. (eds) Financial Cryptography and Data Security. LNCS, Springer. (2013)
33. Luke, S., Ziparo, V.A.: Learn to Behave! Rapid Training of Behavior Automata. In: Proceedings of Adaptive and Learning Agents Workshop at AAMAS. (2010)
34. Lundin, E., Kvarnstrom, H., Jonsson, E.: A Synthetic Fraud Data Generation Methodology. In: Deng R., Bao F., Zhou J., Qing S. (eds) Information and Communications Security, ICICS 2002. LNCS, Springer. (2002)
35. Recio-Garcia, J.A., Gonzalez-Calero, P.A., Diaz-Agudo, B.: Jcolibri2: A framework for building Case-based reasoning systems. Science of Computer Programming (2014)